Emulation-Driven testing of IoT deployments

Context:

The past few years have seen a large growth in the Internet of Things (IoT) industry. Reports predict up to 50 billion of connected devices in 2020 [1]. Typical IoT devices are low power-boards with constrained RAM, CPU and battery life, which communicates over a wireless channel (typically IEEE 802.15.4). The specific characteristics of IoT networks (constrained devices with low duty-cycle, wireless communications in interference-heavy environment) have prompted the development of new stacks of protocols at all OSI layers, such as the IETF standardization of 6LoWPAN+RPL+COAP. Several approaches are used to evaluates these protocols:

- Simulation tools that provide a flexible approach with a variety of network topologies, conditions and simplified logging but require to reimplement every protocol in simulation software such as ns3 [3] or Omnet++ [4] (which have a steep learning-curve and a sometimes unfriendly API).
- Real-life experimentation, for instance by implementing the protocol stack in a modern IoT-OS such as RIOT-OS [2] and deploying it on an IoT network. This approach is costly and not very flexible, as deployment-topologies are often fixed. Furthermore, the constrained memory on typical IoT boards makes logging and event-handling relatively complicated.

In this project, we would like to use the best of both worlds by enabling the large-scale emulation of IoT-networks with RIOT. More specifically, we would like to make it possible to test applications developed in RIOT without an actual testbed and with minimal overhead.

RIOT-OS

RIOT (http://riot-os.org) is an open-source operating system developed specifically for the internet of things. Its goal is to allow fast and efficient development for devices that are not powerful enough to run Linux. As such, it is compatible with numerous sensor boards and has the necessary tools to allow C programmers to write code (almost) out of the box. RIOT code can also be compiled as a Linux process, a feature that makes emulation much easier.

Objective:

The goal of this project is to create a flexible platform to test/evaluate IoT deployments using the RIOT native mode. This platform should allow the emulation of large-scale IoT networks (>103 nodes) on a single (albeit powerful) machine.
Features would include: 802.15.4 link emulation, mobility handling, simple and efficient logging mechanism, etc.

Potential tools:

We have identified certain tools that could be used to complete the project, but this is not an exhaustive list and the candidate is free to explore other options:

- Already existing network simulator/emulator (ns3/omnet++/lurch)
- In-house project: basic bridge in user land based on tap interfaces developed in C providing loss

probability and multicast delivery
- ...

Methodology

This project should be conducted with an engineering-savvy and scientifically sound approach. As a methodology, the student should follow the classical scientific approach:

1. State of the art: evaluate current simulation/emulation methods
2. Draft proposal: design a new emulator and defend the different choices
3. Implementation: build the proposed emulator
4. Evaluation: soundness of the design, accuracy, scalability, etc.

Profile:

- Good knowledge of C/C++ required
- Basic understanding of wireless network specific features (broadcast link, MAC protocols, etc.)

References

[1] D. Evans. The Internet of Things, How the Next Evolution of the Internet is Changing Everything. Cisco White Paper. 2011
[2] Baccelli, Emmanuel, et al. "RIOT OS: Towards an OS for the Internet of Things." *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 2013.
[3] Henderson, Thomas R., et al. "Network simulations with the ns-3 simulator." *SIGCOMM demonstration* 15 (2008): 17.
[4] Varga, András. "The OMNeT++ discrete event simulation system." *Proceedings of the European simulation multiconference (ESM’2001)*. Vol. 9. No. S 185. sn, 2001.

Project supervision: Marcel Enguehard (Cisco Systems, mengueha@cisco.com), Dario Rossi (Telecom ParisTech, dario.rossi@telecom-paristech.fr), Jiazi Yi (Ecole Polytechnique, jiazi.yi@polytechnique.edu)